PCT / IB 03 / 03650
16.09.03

Europäisches Patentamt
European Patent Office
Office européen des brevets



# Bescheinigung Certificate Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

**Patentanmeldung Nr. Patent application No. Demande de brevet n°**

02078544.0

**PRIORITY DOCUMENT**
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 02078544.0
Demande no:

Anmeldetag:
Date of filing: 28.08.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Watermark detection

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H04N5/76

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR

Watermark detection

FIELD OF THE INVENTION

The invention relates to a method and arrangement for detecting a watermark in a media signal.

BACKGROUND OF THE INVENTION

Until very recently the DVD copy-protection community considered watermark-detection for playback-control in a Personal Computer to take place in the DVD-ROM or DVD-rewriter drive. The motivation for this position was that watermark-detection is a permissive technology (i.e. the playback or recording device works with or without watermark detector) as opposed to encryption, which requires a decryptor for the device to function properly. The fragile consensus used to be that DVD-ROM drives would inspect MPEG2-compressed unencrypted DVD-video content on disks for presence of a copy-never or copy-once watermark. If such were the case, playback should be stopped (copy-once or copy-never content should be encrypted at all times). Note that such a drive needs an MPEG2-parser to at least partially decompress content to enable watermark detection. See Fig. 1, which shows schematically a PC system architecture of watermark-detection for playback-control in the DVD-drive.

However, a system with playback-control using a watermark detector in the drive leaves major security holes in an open-architecture PC. One such security hole is that content may be recorded in scrambled form by flipping all bits. Since this is no longer a compliant MPEG2 stream, the parser in the drive will fail and no watermark will be seen. The bit-flip can be undone just before or inside the media-player software. Another such security hole is that content may be compressed not using MPEG2, but MPEG4 (popularized under the name DivX), fractal-compression schemes, Windows Media, Real, etc. Since it is impossible for the DVD-drive to have parsers on board to (partially) decompress all of these (and hackers will invent new codecs to outsmart the drive), the watermark will not be detected. Although (illegal) copies compressed with a codec other than MPEG2 will generally not play on current DVD-video players there is a trend for DVD-video players to support more and more codecs.

Therefore it has already been proposed to place the watermark detector after decompression and just before rendering, i.e. in an MPEG-decoder (hardware) card or in a graphics card. After decompression there is no longer confusion because all content reduces to the unequivocal baseband-format ready for consumption by human eyes. Fig. 2 shows schematically such a PC system-architecture of watermark-detection for playback-control in the graphics card. However, it was considered difficult to enforce MPEG-decoder companies or graphics-card manufacturers to install such watermark detectors. This perception has changed since. An architecture is now being envisaged in which DVD-drives check, on boot-up, whether there is a graphics-card with watermark detector present in the PC. If such a graphics-card with watermark-detector is not present then the drive will not output data. If such a special graphics card is present however, it will output data. When the watermark detector in the graphics card detects a watermark it will try to authenticate to a compliant application which is responsible for rendering the watermarked data. If such authentication is successful, the graphics card continues operation (e.g. a valid DVD-Video is being played back using an authorized application). If it cannot find such a compliant application, the content must have come from some non-authorized source, e.g. an illegally copied disk in the drive is being rendered by some pirate or other non-compliant software. The graphics card will then shut down such output. Fig. 3 schematically shows a sketch of protocols in such a PC architecture to make sure that all components are functioning ensure watermark detection. In this Figure, "Auth." denotes an authentication process or device. Note that the compliant application is certain about the origins of the data which it is rendering because it has also authenticated with the drive. Note also that the architecture is more general. For example, the source may also be an analog capture card, an MPEG-encoder card, or an IEEE-1394 board, rather than a DVD-drive.

It will further be assumed that the watermark detector is located on the output(s) of the graphics card, just before video data is converted to the analog domain or modulated in the digital domain for transmission to A DVI-monitor (see Fig. 2). Although it is architecturally very simple and clean to detect watermarks in the graphics-card, in practice there are a number of problems with this location connected to the enormous amounts of (graphics) data that flow through the graphics part at huge speed, and the fact that multiple streams can be displayed at the same time.

The video coming out of the graphics-card can be at any number of resolutions, see the Table below:

| Standard | Resolution | pixel-clock [MHz] |
|---|---|---|
| VGA | 640 × 480 | 27 |
| XGA | 1024 × 768 | 70 |
| SXGA | 1280 × 1024 | 116 |
| UXGA | 1600 × 1200 | 170 |

Table: Comparison of the resolutions and pixel-clock of some common graphics standards

There are other also other standards supported by some graphics cards, with interpolating resolutions. Note that the pixel-clock for normal baseband watermark detection (in PAL or NTSC) is 13.5 MHz.

Specifically, some of the problems for watermark detection in the graphics card are:

1. The output interface has up to 13× higher data rate (UXGA-mode) compared to normal PAL/NTSC baseband-detection. Baseband detection requires one addition for every pixel, so the adder has to work 13× faster.
2. The data on the outputS is in RGB format whereas most watermark-schemes work with the luminance channel. Conversion from RGB to Y requires 2 additions and 2 multiplications ($(Y/0.587) \equiv 0.509\,R + G + 0.194\,B$, where $0 \leq R,G,B < 1$). This is very costly, especially at high data rates.
3. The potential range of scales that the detector needs to deal with is very large. One of the highest scales still preserving visual quality is to display the movie full-screen on the monitor (blow up to 1600×1200 or even more). Roughly the lowest scale is when the video is reduced to 352×200 (a popular format for downloading movies from the internet). The scale-range horizontally is thus 0.5...2.2 and vertically 0.4...2.5, whereas currently available watermark detectors are designed to deal with scales in the range 0.5...1.5.
4. As shown in Fig. 2, there are usually multiple outputs on the graphics card, currently VGA-out and TV-out (for displaying a DVD-movie rendered on a PC on a living-room TV). Recently the digital DVI-interface has been added to this palette. Because all these outputs can be controlled independently (i.e. display different data), naively the number of detectors should equal to the number of outputs, which constitutes a significant cost-burden.

5. In the architecture of Figs. 2 and 3, a hacker may perform the following hack: (s)he copies illegal content which (s)he wants to watch from a DVD+R to the HDD, without rendering. Then (s)he plays any valid protected DVD-video from the DVD-drive with a compliant application in one window, while the illegal material is rendered by a non-compliant application in another window. The watermark detector will find a watermark but that is (it thinks) consistent with the original movie in the drive. Thus the illegal material is not caught. It is even possible to abuse a compliant application: the illegal content from the HDD can re-encrypted with CSS (which has been hacked), thus disguising it as valid content. This ReCSS-ed content is thus accepted by the compliant player, and after watermark detection in the graphics card, this application will vouch for it.

6. If a watermark is found by the compliant graphics card but no compliant application which produced the (watermarked) video can be found (or the drive cannot confirm that the source is indeed a DVD-Video disk), what should the graphics card do to stop the display of the illegal movie?

7. In the architecture of Figs. 2 and 3, a hacker may perform the following hack: he inserts a second non-compliant graphics card into the PC. He allows the drive to authenticate to the graphics card (using a hacked driver), while he uses the non-compliant card to playback illegal material from the drive. A second hack scenario is when he only inserts a non-compliant graphics card into his PC but connects the PC via a network (home network or internet) to another PC with a compliant graphics card. After authenticating the drive with the remote compliant graphics-card, illegal content is displayed on the on-board non-compliant graphics card. A third hack scenario is where there is a compliant DVD-drive and a compliant graphics card with watermark detector in a single PC: after authentication the hacker streams the data from illegal disk in the drive to a non-compliant application running on another PC with a non-compliant graphics card somewhere in the network.

8. A hacker may open up the graphics card and learn the secret behind the authentication (like what happened for DVD-Video), and publish it on the net. From then on, a compliant graphics card can always be impersonated by a non-compliant graphics card plus special software tool using this secret. Of course the hacker may also steal the watermark-secret, but removing a watermark from a movie is computationally much more work intensive than faking an authentication.

OBJECT AND SUMMARY OF THE INVENTION

In this section we provide a set of possible solutions to the problems 1 to 8 mentioned above. The same numbering is adhered to.

1. Trivially: the pixel-data is sub-sampled in space and time (by skipping): e.g. the only information used for detection is line 1 from frame 1, line 2 from frame 2 etc. Alternatively, only a part of the images is being watermark detected: see also solution 3 below.
2. The multiplications to convert from RGB to Y can be avoided by approximating Y, e.g. $Y \approx 0.25\,R + 0.5\,G + 0.125\,B = R/4 + G/2 + B/8$ which can be implemented with only arithmetic shifts. Furthermore, to economize on additions, we can approximate further $Y \approx G$ (because G is dominant).
   Alternatively, the 3 primary colors R,G,B can be accumulated / folded / processed separately, using 3 separate fold-buffers. Then after folding, RGB can be converted to Y off-line, instead of on-the-fly (this is much like the way in which known watermark detectors accumulate DCT-coefficients on which finally an IDCT is performed rather than performing an IDCT on-the-fly and then accumulate the resulting pixels). This procedure takes 3 times more memory but it can usually still be neglected w.r.t. the amounts of video-memory used for other purposes. Memory bandwidth can be a problem though, because 3 times as much data must be transported to memory.
3. It is usually quite easy to distinguish video from all the other information on the desktop, because real-time video contains many more changes. This could lead to the following solutions:
   a. the watermark detector stores the (thresholded) changes with respect to the previous frame. It tries to fit a bounding box around (rectangular area that includes) areas with significant change. That bounding box will be considered a window on which normal watermark detection (scale detection followed by payload detection) is performed. In other words, whereas before we had only scale detection and payload detection, we have now added "area-of-interest-detection". Fig. 4 shoes schematically the structure of graphics-card watermark-detection according to this solution.
   b. same as a., but the change-detection is performed on a sub-sampled video-frame to conserve storage space.

c. same as a, but the change-detection is performed "block-for-block" (e.g. first try to find the change-areas in the top-left corner, then in the top-right corner etc.).

It is well known from the literature how, starting from an area of activity one can determine the tightest possible bounding-box (red rectangle in figure above) including such a point. If the content in the area-of-interest is now upsampled or downsampled to the normal 720×480 or 720×576 format, and supplied to a normal baseband watermark detector, it is very likely that the content is now processed at a scale sufficiently close to 1.0.

4. Time-multiplex the watermark detector onto the different outputs: i.e. first detect for a fixed amount of time on output 1, then on output 2 etc.. It is also possible to check all outputs simultaneously.

5. When the detector has found watermarked content, which (through authentication) can be traced back to a compliant application or drive, the watermark detector should continue to search other areas of the display, to search illegal content with watermarks. In practice one could implement this by starting the bounding-box search of solution 3 at a random point on the display, to avoid ending up with the same bounding-box all the time. As an alternative the graphics card may notify the drive of the watermark-payload using the authenticated channel set up at boot-time. The drive can verify from the disk whether this watermark-payload is commensurate with this disk. If not, some other source of copied material must exist. Note that for this method to work, the watermark-payload needs to be stored on the disk in a manner that it cannot be retrieved by a hacker, e.g. in some currently unused sector in the lead-in area. This does not add cost to the drive.

6. Using the information from change detection in solution 3, blank out, fog over, or otherwise destroy the viewing pleasure of the boxed area in which the watermark was detected. Alternatively, scroll a message across the whole image indicating the detection of a watermark in a non-authenticated stream.

7. The operating system and the BIOS are the only entities in the PC which have reliable knowledge about the plug-in card configuration of the PC. A solution for the first hack-scenario is for the BIOS or OS to prohibit combinations of compliant and non-compliant graphics cards in a PC (for security reasons). A solution for the second hack-scenario is for OS and BIOS to disallow authentication with graphics cards across the network. A way to implement this would be for the OS to query the drive which graphics card it authenticated with and to check that the device is indeed on board. This

obviously requires a secure OS. If it is a market requirement that playback from a remote DVD-drive in a home network should be allowed, the second scenario hack of problem 7 cannot be prevented. Another solution is for the OS to prohibit combinations of compliant drive and non-compliant graphics card in the same box.

8. The authentication-scheme for the watermark detector should be endowed with a revocation scheme as championed by e.g. Philip's Sapphire architecture. If a hacker opens up a graphics card, such card can be revoked by (can no longer communicate with) future drives, OS-es, media etc.

Note that, as mentioned before, the DVD-drive could be replaced by any other compliant source of data in these solutions.

The invention can be summarized as follows. Watermark-detection in a graphics card for the purpose of copy-protection in a PC, has recently started to draw a lot of attention in standardization. However, detection in a graphics card has problems completely different from the formerly considered detection in the DVD-drive, having to do with high data-rates, large scale-ranges and presence of multiple video-streams in the display area. This disclosure discusses a number of approaches to confront these problems.

CLAIMS

5

1. A method and arrangement for detecting a watermark in a multimedia signal as described hereinbefore.

Abstract

Watermark-detection in a graphics card for the purpose of copy-protection in a PC, has recently started to draw a lot of attention in standardization. However, detection in a graphics card has problems completely different from the formerly considered detection in the DVD-drive, having to do with high data-rates, large scale-ranges and presence of multiple video-streams in the display area. This disclosure discusses a number of approaches to confront these problems.

Fig. 2

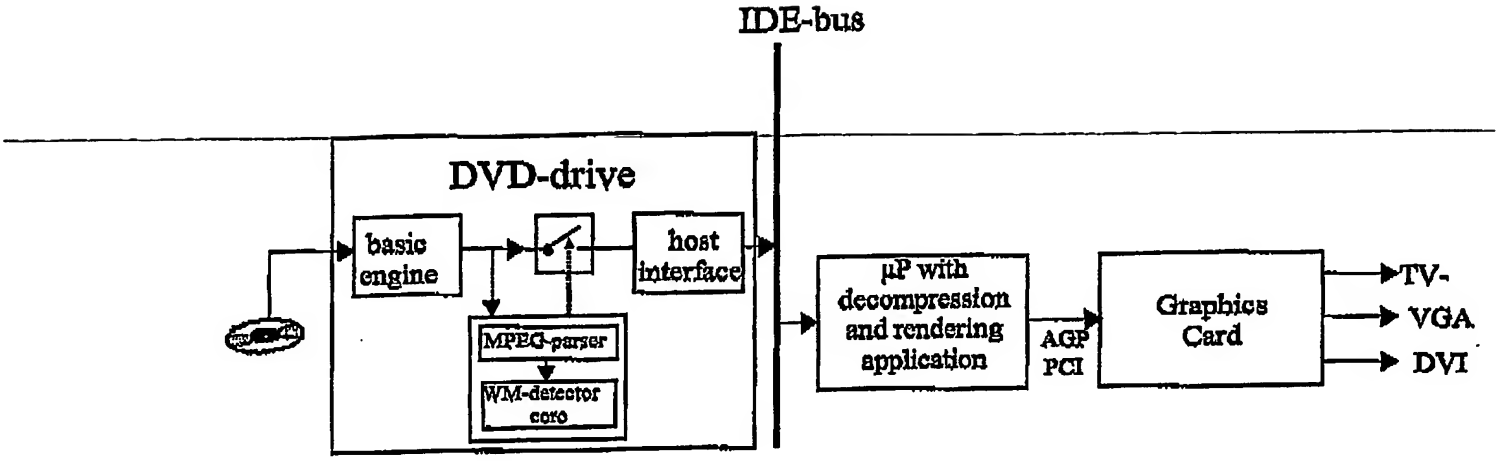
IDE-bus
DVD-drive
basic engine
host interface
MPEG-parser
WM-detector core
µP with decompression and rendering application
AGP PCI
Graphics Card
TV-
VGA
DVI

FIG. 1

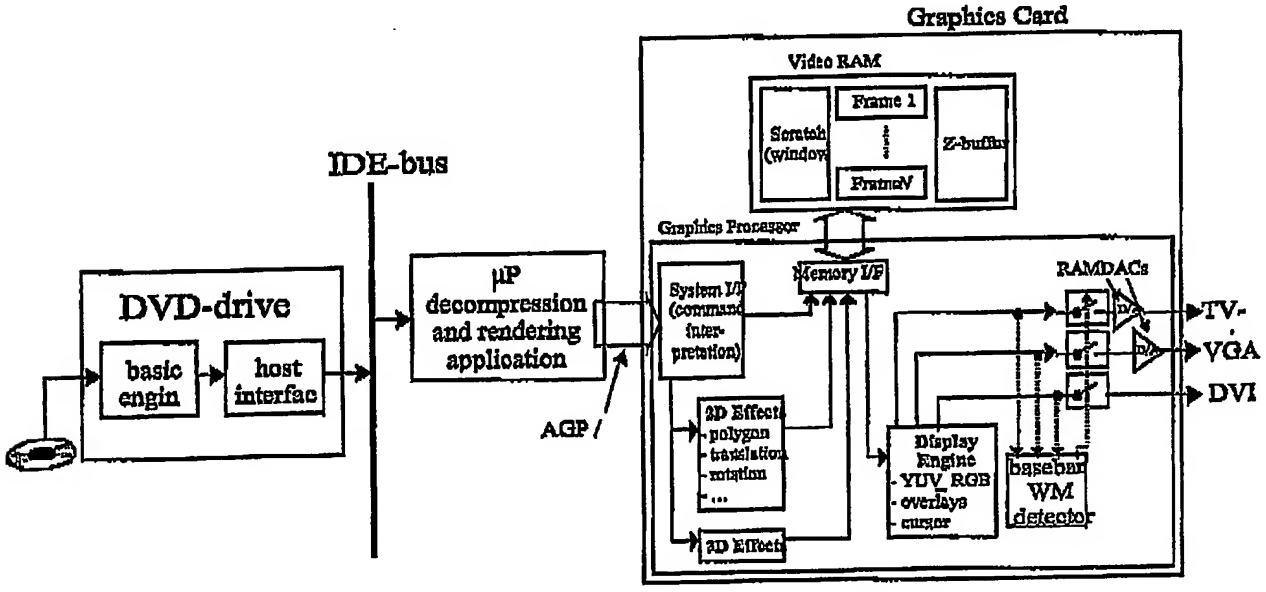
Graphics Card
Video RAM
Frame 1
FrameN
Graphics Processor
Memory I/F
RAMDACs
IDE-bus
DVD-drive
basic engin
host interfac
µP decompression and rendering application
AGP
2D Effect
- polygon
- translation
- rotation
- ...
3D Effect
Display Engine
- YUV_RGB
- overlays
- cursor
baseban WM detector
TV-
VGA
DVI

FIG. 2

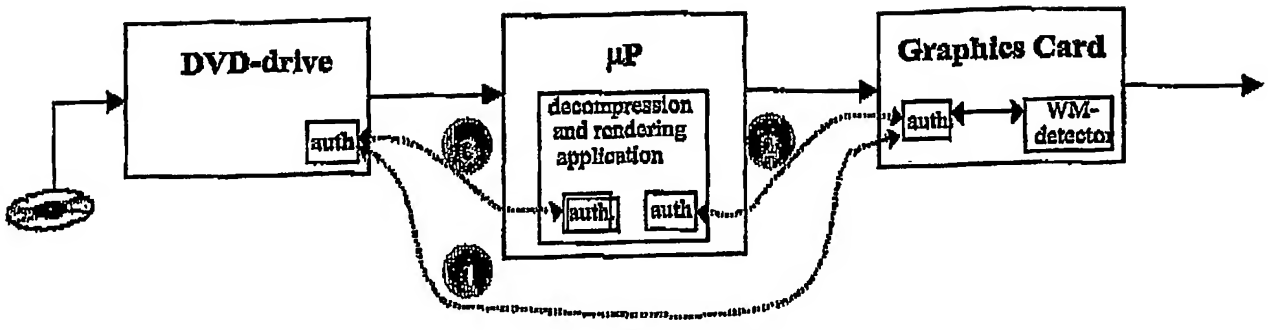
DVD-drive
auth
µP
decompression and rendering application
auth
auth
Graphics Card
auth
WM-detector
1
2
3

FIG. 3

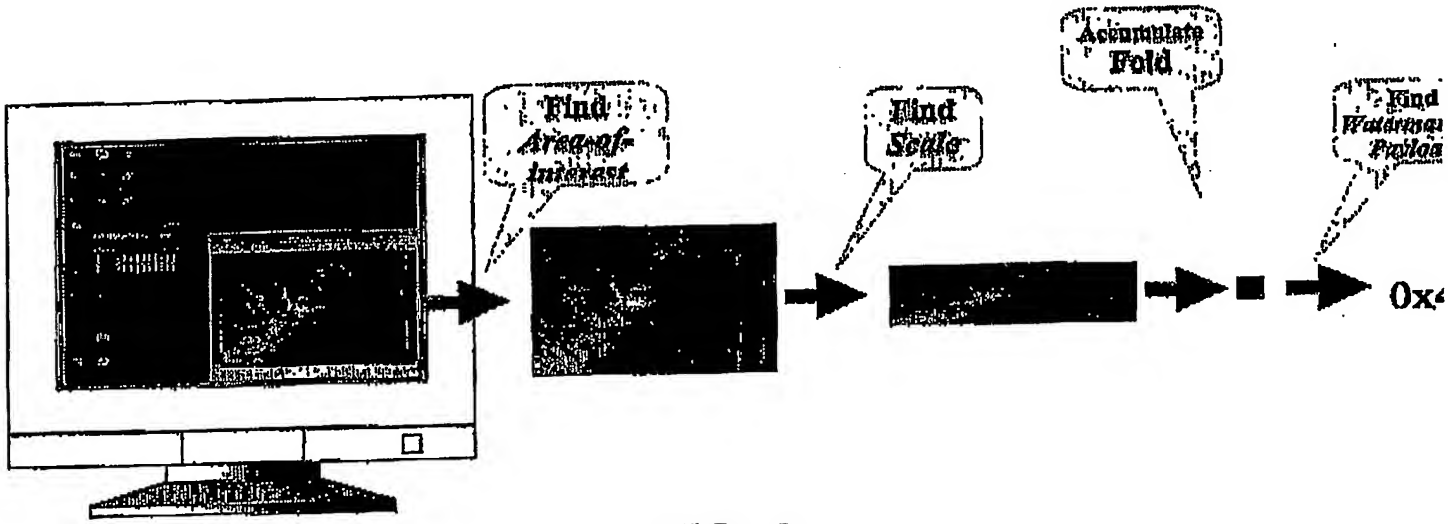
Find Area-of-interest
Find Scale
Accumulate Fold
Find
0x

FIG. 4